Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital
IMD0036 – Sistemas Operacionais

Gerência de Arquivos: Parte 2

Prof. Gustavo Girão

girao@imd.ufrn.br

Roteiro

- Estrutura de Discos
- Gerenciamento de Espaço em Disco
- Desempenho do Sistema de Arquivos
- RAID

ESTRUTURA DE DISCOS

Estrutura de Discos

- Um disco é dividido em ***trilhas***
- O número de trilhas é particular de cada dispositivo
- Cada trilha dividida em ***setores***
- Setor a menor unidade de informação que pode ser lida ou escrita em um disco
  - E os blocos??
    - São unidades menores do que setores, porém não são a unidade mais básica de **transferência** de um disco

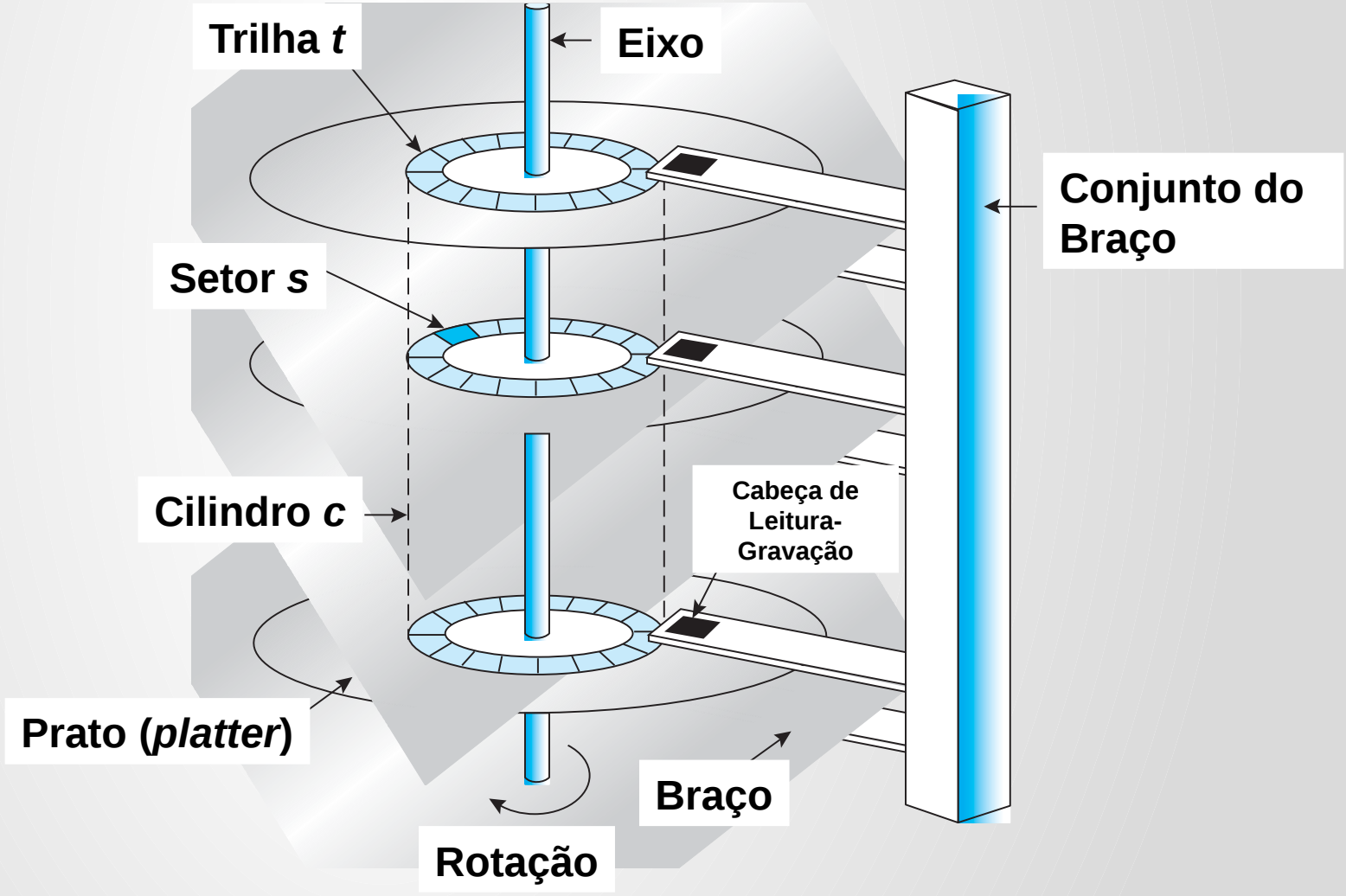
Estrutura de Discos

- Acesso a um setor:
  - informar ***prato***, **trilha** e **setor** desejado
  - as cabeças de leitura e gravação são deslocadas até a trilha correta (*tempo de seek*),
  - o cabeçote da face correta é selecionado eletronicamente
  - esperar até o setor desejado passar por baixo do cabeçote (*tempo de latência*)

Estrutura de Discos

- Normalmente, uma unidade de disco rígido é formada por vários discos superpostos (no máximo 8 discos, com a tecnologia atual)
- ***Cilindro***: é formado pelas trilhas que estão na mesma posição, porém em diferentes faces (no caso de 8 discos superpostos cada cilindro é formado por 16 trilhas).
- Não é necessário deslocar as cabeças para acessar trilhas de um mesmo cilindro

Geometria do Disco



Desempenho de Discos

- **Latência de acesso** = tempo médio de acesso = Tempo médio de Seek + Latência Média
  - Discos rápidos: 3ms+2ms = 5ms
  - Discos lentos = 9ms + 6ms = 15ms
- **Tempo médio de acesso a E/S** = tempo médio de acesso + (quantidade a ser transferida / taxa de transferencia) + overhead do controlador

Primeiro Disco Comercial

1956
IBM RAMDAC computer incluía o sistema de armazenamento em disco **IBM Model 350** (5 MB!)

5M (7 bit) caracteres
50 x 24” pratos
Tempo de acesso = < 1 second

Discos de estado sólido

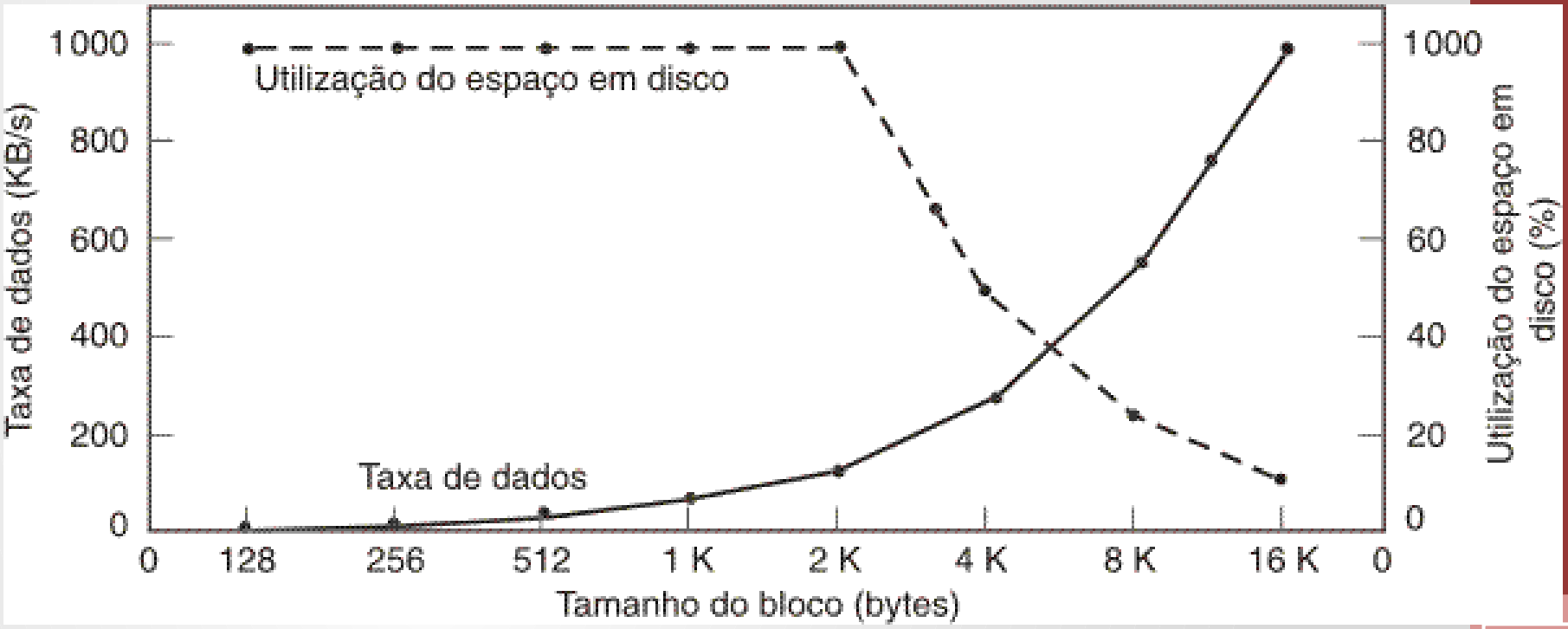
- **SSDs (Solid-State Disks)**
  - Memória não-volátil utilizada como um disco rígido
    - Muitas variações tecnolóicas
  - Pode ser mais confiável do que HDDs
  - Mais cara do que HDDs (preço por MB)
  - Pode ter um tempo de vida mais curto
  - Menor capacidade
  - Muito mais rápida
  - Não existem partes mecanicas: não há tempo de busca ou latência rotacional

GERENCIAMENTO DE ESPAÇO EM DISCO

Administração de espaço

- Questão importante é como o espaço em disco é administrado
  - Sequência contígua de bytes
  - Sequência de blocos (não necessáriamente contínuos)
- Tamanho do bloco
  - Candidatos
    - Setor, trilha, cilindro
    - Desperdício de espaço vs. Taxa de transferencia de dados (latencia de movimento de braço + rotação + taxa de transferencia de cada bloco)
  - Tamanho dos arquivos
    - Unix (2055) mediana do tamanho dos programas é 2k
    - Windows em sistemas científicos (Cornell), idem

Administração de espaço



Administração de espaço

- Espaço livre
  - Blocos com lista ligada de blocos livres
    - Ultima entrada é próximo bloco da lista
    - Apenas 1bloco na memória de cada vez
    - simples implementação
    - Disco de 1Tb, 32 bits para número do setor, setor de 1Kb, lista livre ocupa até 4Gb
  - Vetor de bits
    - 1 bit por bloco do disco
    - Tamanho fixo
    - Muito compacto, 1 bit por bloco
      - Bloco de 1k contém indexação de 8 mil blocos (8Mb)
      - Disco de 8Tb indexado com 125Mb de lista
    - Implementação de busca mais complexa

Compartilhamento de Arquivos

- Quando usuários diferentes compartilham arquivo, é conveniente se arquivo aparece simultaneamente em diretórios diferentes
- Se é possível acessar arquivos por caminhos diferentes, estrutura de diretórios vira DAG (não mais árvore)
- Novas conexões são chamadas de “links”
- 2 maneiras
  - Hard links – entradas em diretórios distintos compartilham descritor do arquivo
    - Assim lista de blocos não faz parte da entrada do diretório
    - UNIX
  - Links simbólicos – tipo especial de arquivo contendo caminho com nome da localização real do arquivo

Compartilhamento de Arquivos

- Problemas
  - Links simbólicos
    - Sobrecarga de acesso – apenas na abertura
    - Entradas órfãs de arquivos eliminados
  - Hard links
    - Remoção de arquivo deve cuidar para que arquivo real seja removido apenas quando não há mais “links”(I-node tem contador de links)
    - Usuário original contuará sendo cobrado pelo espaço
    - Arquivo removido pelo dono continua sendo usado
    - Ciclos
  - Geral
    - Back-ups devem cuidar para não duplicar trabalho e reconsitituição deve manter unicidade

Confiabilidade do Sistema de Arquivos

- Arquivos mais valiosos que o hardware
- Localização dos setores ruins do disco
  - Hardware – setor do disco contém lista dos blocos ruins com lista dos blocos substitutos. Controlador usa substitutos automaticamente
  - Software – sistema de aquivos constrói arquivo contendo blocos ruins
- Atualização atômica
  - Garantia que atualização parcial nunca acontece, mantendo arquivos sempre consistentes
  - Se atualização falhar antes de comletar, basta rodar de novo
  - Discos com atualização atômica tolerante a falhas são implementados usando **2 ou mais** discos físicos para **um** lógico.

RAID

- *Redundant Array of Independent Disks*
- Esquemas para operação de vários discos como se fosse apenas 1, criando um conceito de **disco virtual**
- Original “RAID5” 1978 (Norman Ken Ogushi)
- Implementação por Hardware (placa) ou software
- O objetivo principal é aumentar a **performance/vazão** do disco
  - Também é utilizado para aumentar a **confiabilidade** por meio de redundância
- Foram criados diferentes padrões de formação
  - Chamados de níveis (RAID nível 0, RAID nível 1 , etc...)

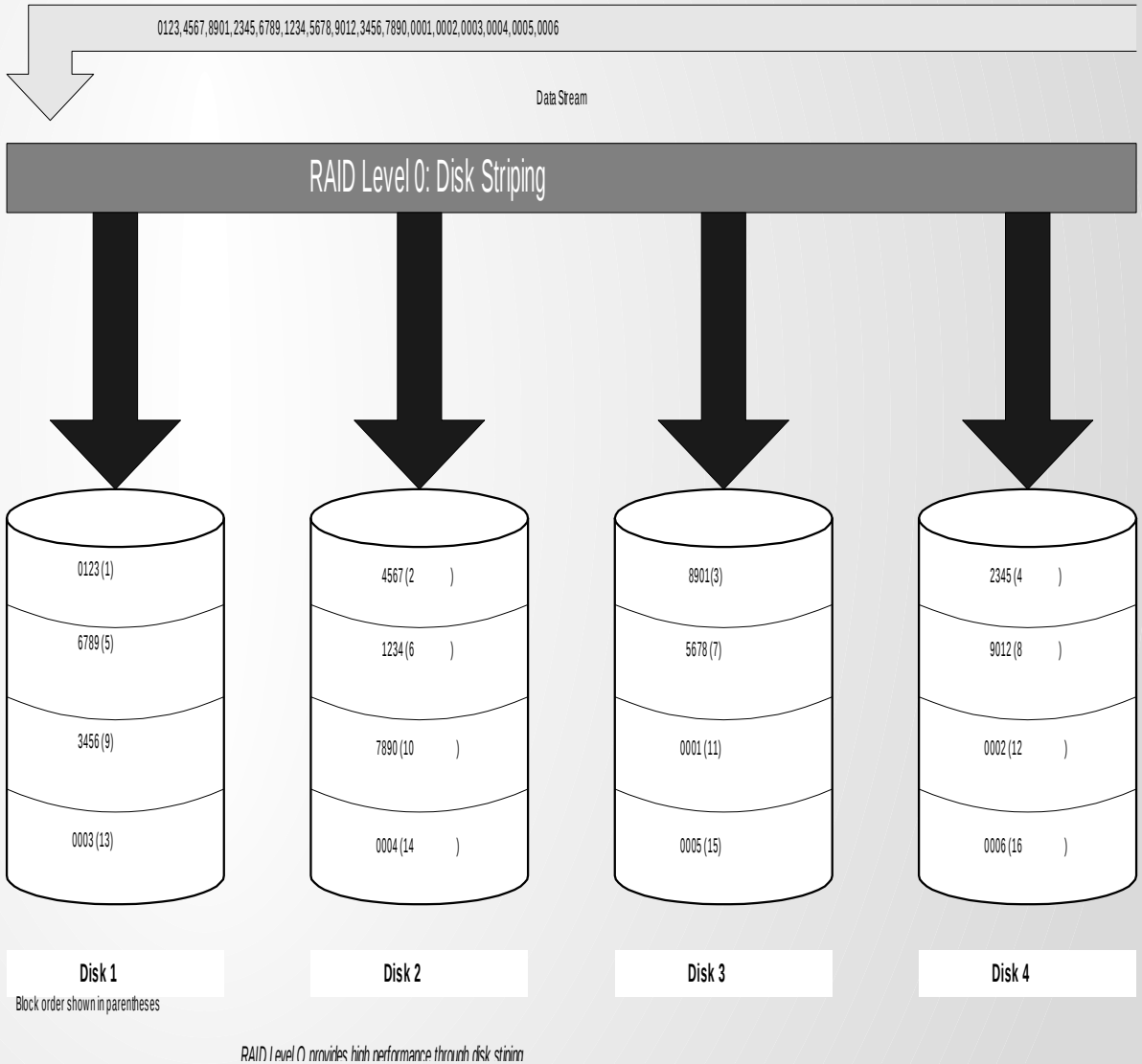
RAID

- **Disponibilidade**
  - Aparece no servidor como um único Grande Disco Rígido .
  - Sistema Operacional totalmente independente.
  - Dispositivo com alta confiabilidade e redundância.
- **Aumento da Disponibilidade dos Dados**
  - Rápido acesso aos dados (multiple drives).
  - Segurança oferecida pela redundância/paridade.
- **Gerenciamento Simplificado**
  - Partição de um único Drive para gerenciamento/partição.
- **Componentes Hot-Swap**
  - Tanto para um ou vários drives.
  - Fontes de Alimentação e Ventilação, também.

RAID nível 0

- Uso de múltiplos Discos para a formação de um único Disco lógico.
- Alta performance na Escrita e Leitura (Write and Read) performance relacionada com o aumento da quantidade de Discos.
- Os Discos Rígidos são distribuídos utilizando-se uma tamanho definido de “**stripe**” durante a configuração
  - Deve ser otimizado em conjunto com o Sistema Operacional para uma performance otimizada
- As pequenas solicitações que possuem o mesmo tamanho de “**stripe**” são transmitidas a um único Disco Rígido, as solicitações maiores são divididas e transmitidas a múltiplos Discos Rígidos em paralelo
- A capacidade é a soma do número de discos no “array”
- Não proporciona proteção contra falhas de hardware, somente performance

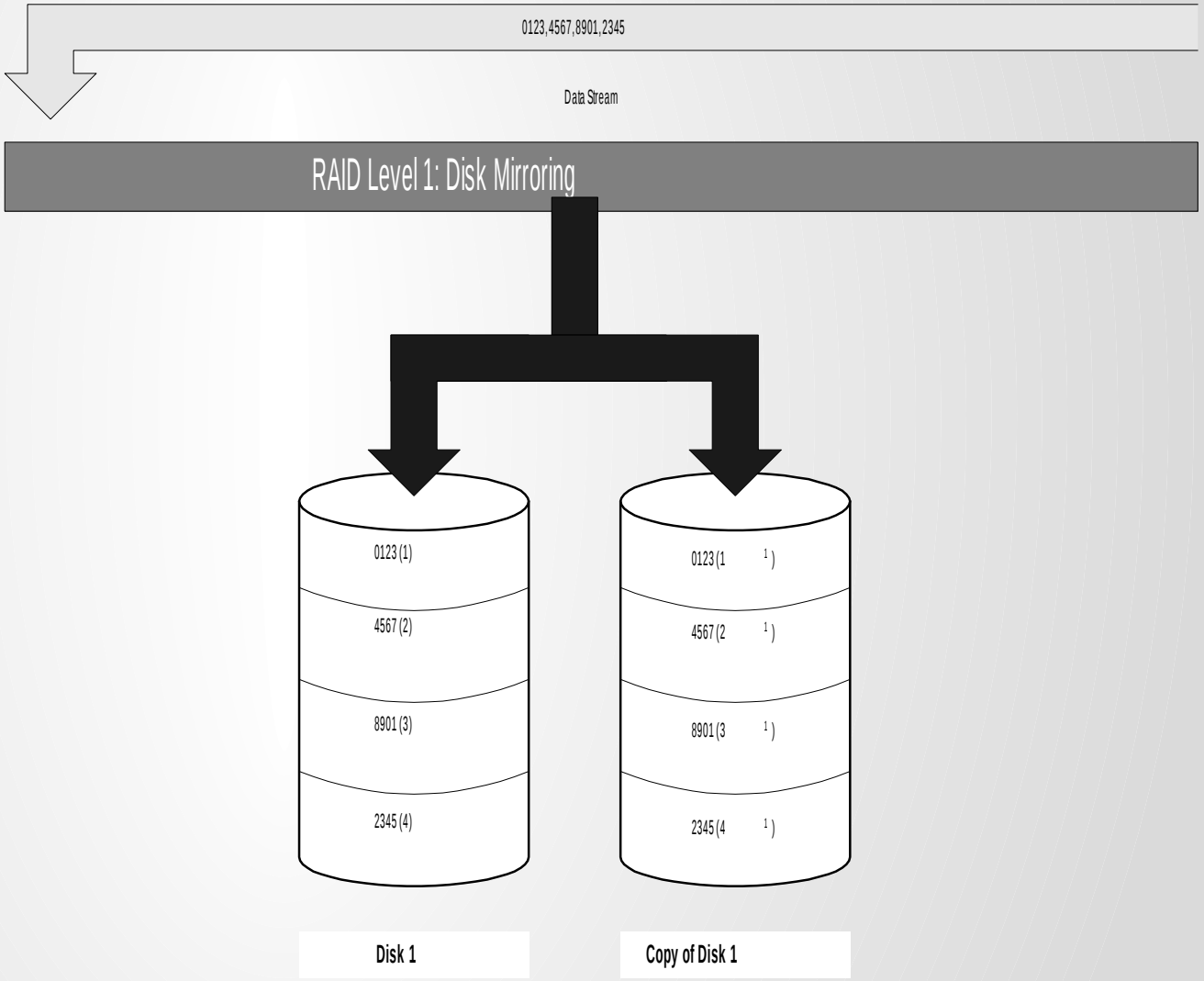
RAID nível 0


0123,4567,8901,2345,6789,1234,5678,9012,3456,7890,0001,0002,0003,0004,0005,0006
Data Stream
RAID Level 0: Disk Striping
0123 (1)
6789 (5)
3456 (9)
0003 (13)
4567 (2)
1234 (6)
7890 (10)
0004 (14)
8901(3)
5678 (7)
0001 (11)
0005 (15)
2345 (4)
9012 (8)
0002 (12)
0006 (16)
Disk 1
Disk 2
Disk 3
Disk 4
Block order shown in parentheses
RAID Level 0 provides high performance through disk striping

RAID nível 1 (espelhamento)

- **O RAID 1 proporciona alto nível de tolerância a falhas**
  - Cada solicitação de I/O é espelhada em um segundo Disco Rígido
- **O RAID 1 trabalha com múltiplos de dois Discos Rígidos** - o set primário e o set espelhado – dobrando também o custo por GB da solução
  - Proporciona o mesmo throughput durante a escrita
  - Oferece mais performance durante a leitura (read) – **Paralelismo**
- **Oferece proteção contra falhas nos discos com taxa de 1 para 1**
  - Assim que um disco falha, automaticamente o espelho assume, porém caso ocorra a falha no espelho não existe mais segurança
  - Pode proteger contra desastre naturais/físicos porém o espelho deve estar instalado remotamente (em outro local físico) acarretando em um custo adicional $$.
- Implementação de Alto Custo
  - Controladora RAID +
  - Custo por GB de chega a ser o dobro se compararmos com uma unidade simples.

RAID nível 1

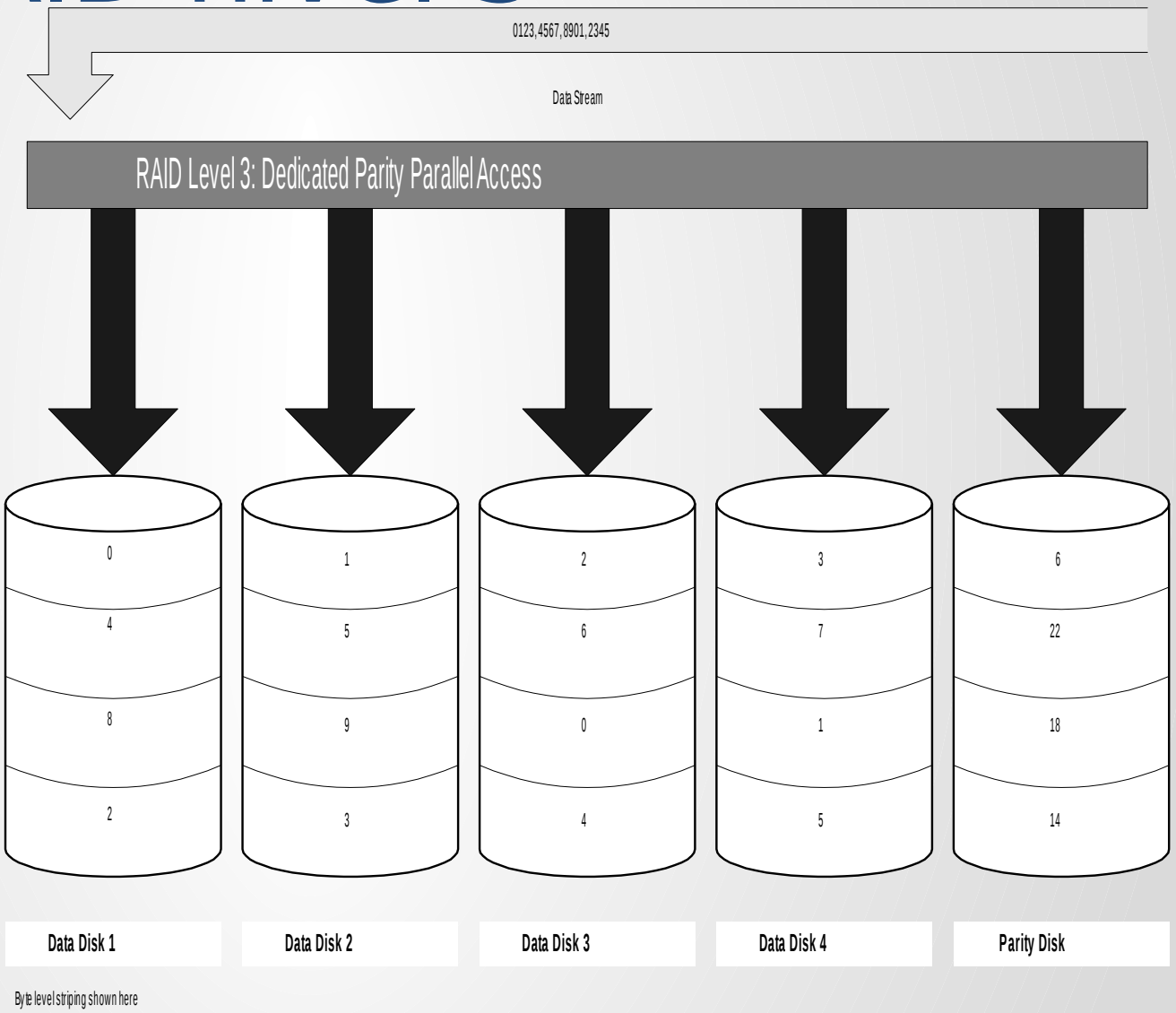


RAID Level 1 provides fully redundant disk mirroring

RAID níveis 2, 3 e 4

- **RAID nível 2**
  - Similar ao nível 1, porém utiliza Error Correction Code para verificar erros nos dados
  - Bits de paridade são utilizados para determinar se houve erro
  - Verifica-se quantos bits dentro de um byte são iguais a 1: se for **par**, o bit de paridade é **0**, se for **ímpar**, o bit de paridade é **1**
    - **0110 0101** = bit de paridade igual a **0**
    - **1100 0111** = bit de paridade igual a **1**
  - Se a paridade for inconsistente, o dado está errado
- **RAID nível 3**
  - Melhora o nível 2: considera qeu os controladores de disco são capazes de identificar o erro.
  - Portanto o armazenamento extra é utilizado para bits de **correção** do erro
  - Na prática o nível 2 **não** é utilizado e sim o 3 ou 4
- **RAID nível 4**
  - Similar ao RAID nível 3
  - Utiliza um granularidade diferente para a paridade: blocos

RAID nível 3



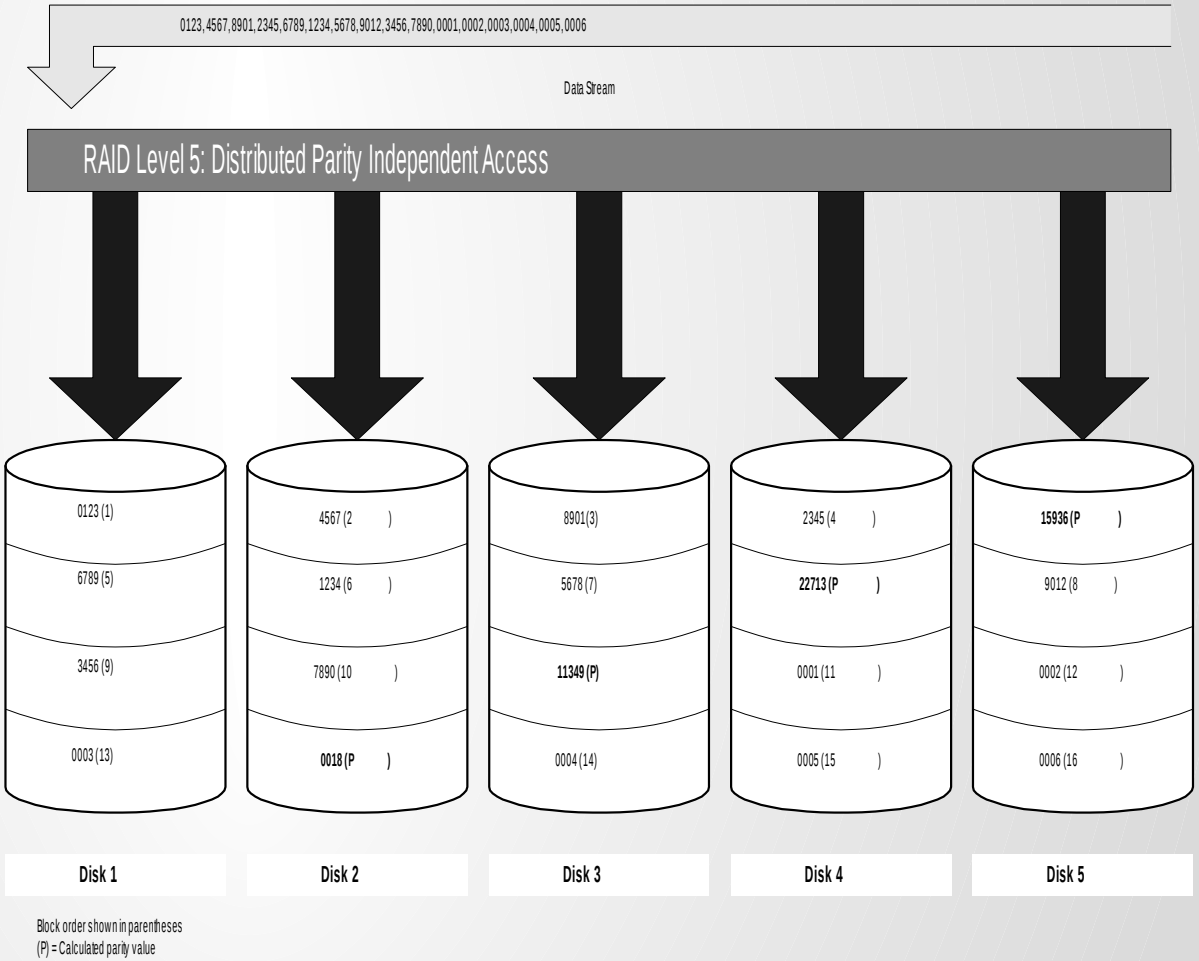
RAID Level 3 uses separate parity disk

RAID nível 5

- O uso de mais discos possibilita o aumento da performance RAID 5, é geralmente utilizado em ambientes OLTP
- Proteção contra falha de Disco com razão de 1 para vários
  - Qualquer disco rígido (1) pode falhar e mesmo assim o sistema permanece intacto
  - Não oferece proteção contra desastre físicos
  - Não protege contra vírus digital ou acidentes/delete intencional, já que os dados são protegidos por um esquema de paridade ECC (com esquema e atualização em tempo real)
- Implementação de Alto Custo
  - Capacidade com perda pela Paridade, o crescimento dos dados aumenta o custo por GB
  - Custo da Controladora RAID +
  - Implementado como um sistema RAID

    (gabinete stand alone)

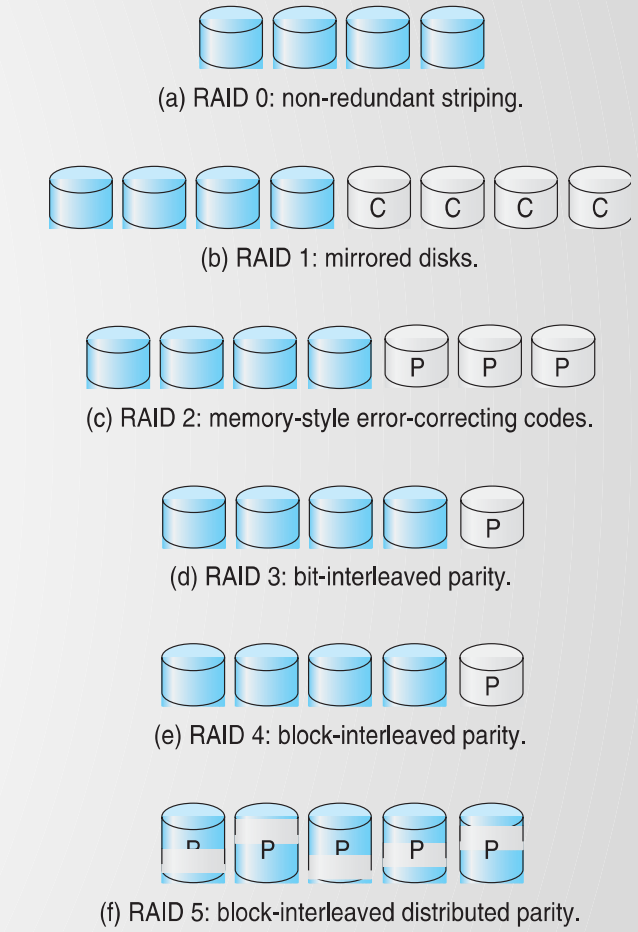
RAID nível 5



RAID Level 5 distributes parity across all drives

Resumo dos níveis de RAID

- Exemplo para uma quantidade dados úteis equivalente a **quatro** discos



Confiabilidade do Sistema de Arquivos

- Controle de Concorrência
  - Unix tradicional – requisição de leituras e escritas executadas na ordem em que chegam
  - Pode gerar problemas quando exclusão múdua é necessária
  - Solução mais comum são travas (“locks”)
    - 1 por arquivo
  - Mesmos problemas que exclusão mútua (ex. Usuário “pifa”)
  - Diferença importante – ênfase nos dados (ao invés de codigo)

Confiabilidade do Sistema de Arquivos

- Transações
  - Travamento automático + atualização atômica
  - Begin_transaction
    - Atualizações
  - End_transaction
  - Nada acontece até “end_transaction”
- NTFS (http://msdn.microsoft.com/en-us/magazine/cc163388.aspx)

Cópias de Segurança

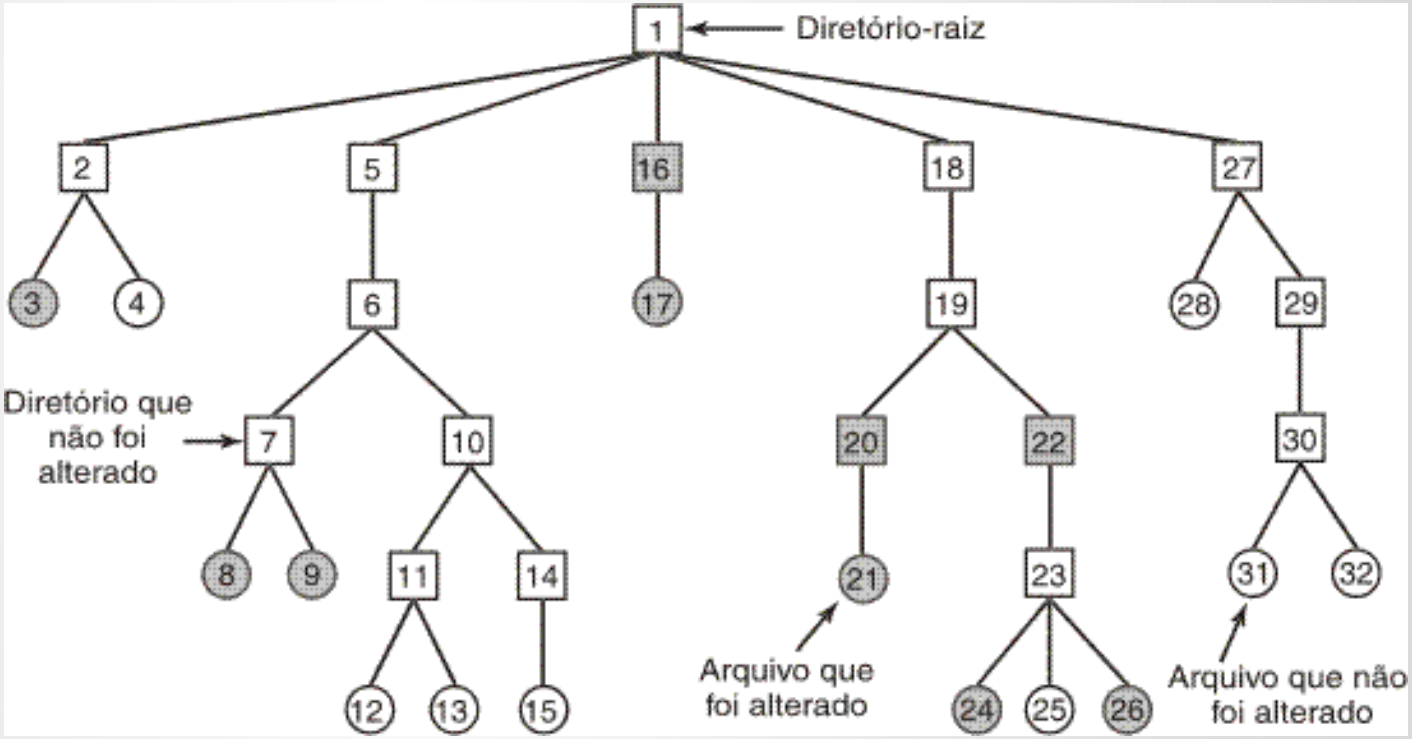
- Dois objetivos
  - Recuperação de desastres
  - Recuperação de erros
- Potencialmente lentos e alto uso de espaço
- O que recuperar?
  - Dump Físico vs Dump Lógico
- Dump físico
  - Disco inteiro é copiado
  - Simples mas custoso
  - Cuidado com blocos inválidos
    - Se mantidos pelo hardware, ok
    - Se mantidos pelo SO, programa de backup deve ter acesso a estruturas e evitar copiá-los

Cópias de Segurança

- Dump lógico
  - Muito sistemas não fazem backup de executáveis, arquivos temporários, arquivos especiais (/dev/ é até perigoso)
  - Em geral é interessante especificar diretórios a serem guardados
  - Começa em um ou mais diretórios especificadfos e recursivamente percorre a estruturq de diretórios salvando os itens encontrados
  - não copiar arquivos especiais (pipes,etc.)

- Dumps incrementais
  - Não tem sentido fazer novo backup de arquivos não mudados
  - Dump completo + incrementos (e.g. Mês + dia)
  - Complica recuperação
  - Mesmo diretórios não modificados devem ser salvos para facilitar recuperação
    - Domingo - Full dump
    - Segunda - Backup de /usr/local/ndr2/xpto/arq1.txt
    - Terça – quero remover /usr/local/ndr2
    - Quarta – como recuperar /usr/local/ndr2/xpto/arq1.txt? -> preciso recriar ndr2 e xpto

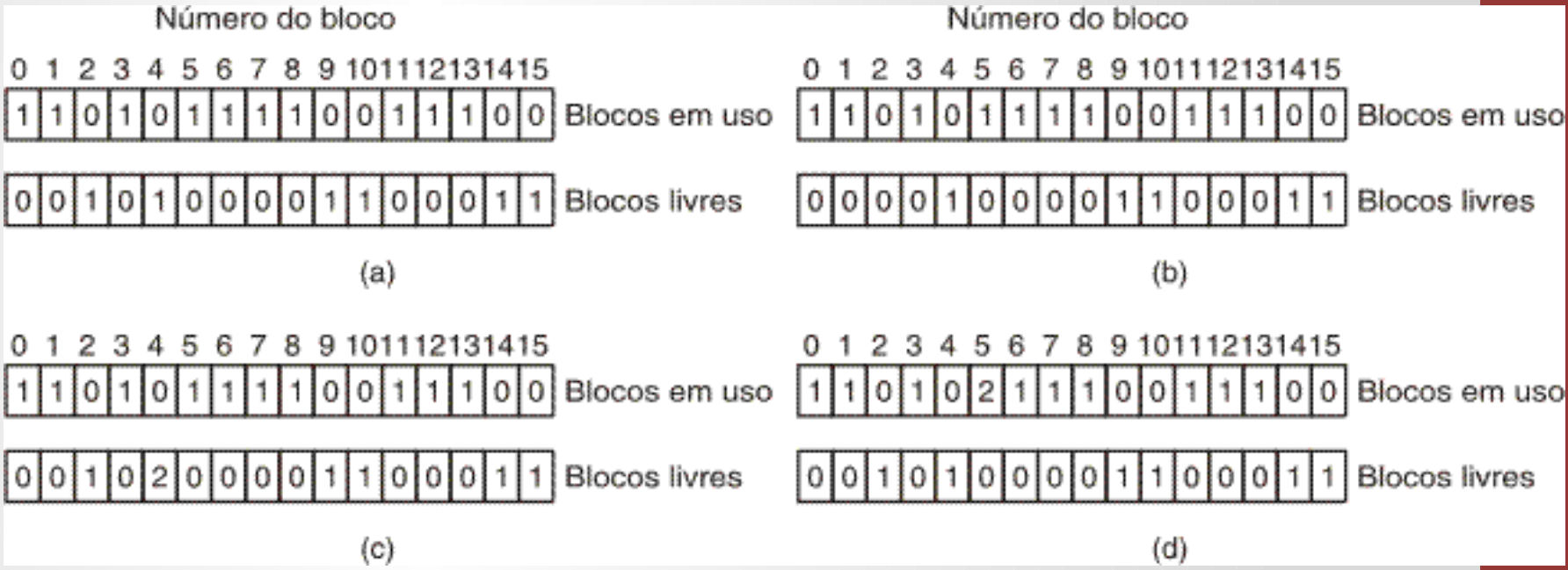
Cópias de Segurança



Confiabilidade do Sistema de Arquivos

- Outras questões
  - Unix pode ter “buracos” nos arquivos
    - “open, write, seek, write”,
    - core dumps tem espaço entre código e pilha
    - não queremos “buracos” preenchidos na recuperação.
- Cuidado com links para evitar duplicação (e loops)

Confiabilidade do Sistema de Arquivos



Confiabilidade do Sistema de Arquivos

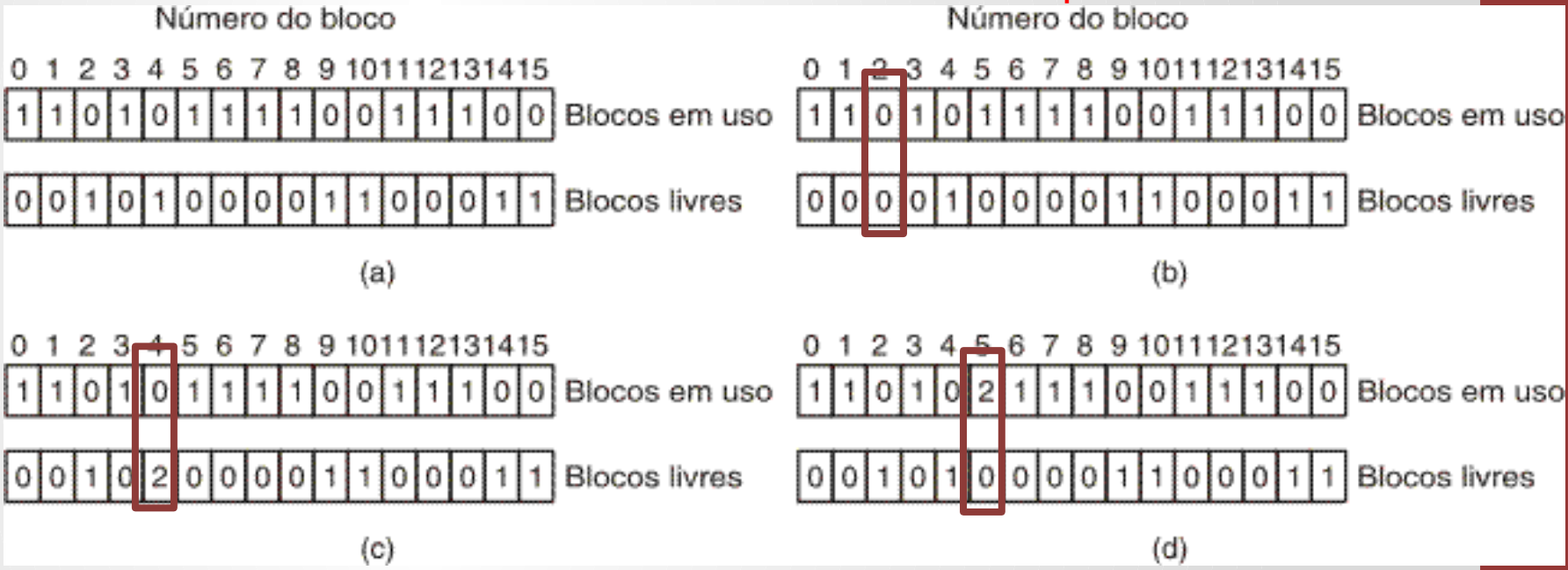
- Sistemas de arquivos lêem/criam blocos, podem modificá-los e depois são salvos
- E se programa morre antes dos blocos serem salvos?
  - Mais grave se bloco é i-node
- Maioria dos SOs têm programas para verificar consistência do sistema de arquivos.
  - Unix – fsck
  - Windows – chkdsk

Consistência: fsck

- Consistência de Blocos
  - Duas tabelas com contadores para cada bloco, inicializados com zero
    - Quantas vezes um bloco está presente em um arquivo
    - Quantas vezes um bloco está presente na lista livre
  - Programa lê todos os i-nodes e percorre lista de blocos
    - Toda vez que bloco é encontrado atualiza primeira tabela
  - Programa percorre lista livre
    - Toda vez que bloco é encontrado atualiza segunda tabela
  - Blocos bons => (1,0) ou (0,1)
  - Missing block (0,0) => adicionado à lista livre
  - Bloco com mais de uma ocorrencia em lista livre (0,n) – reconstrói a lista livre
  - Blocos presentes em mais de um arquivo (n,0)
    - Situação mas grave – deve gerar depois (n,m) ou (0,m)
      - Faz cópia do bloco e insere a cópia em um dos arquivos
      - Quase com certeza um bloco está corrompido

Confiabilidade do Sistema de Arquivos

Bloco desaparecido!



Bloco duplicado !

Bloco duplicado !

Consistência: fsck

- Consistência de arquivos e diretórios
  - Tabela de contadores, um por arquivo
  - Percorre árvore de diretórios
    - Incrementa contador toda vez que um i-node é encontrado (lembrem-se que arquivos podem ser apontados por mais de um diretório por “hard links”.
  - Compara contadores com número de links nos i-nodes respectivos
  - Link count muito alto
    - Sistema não atualizou contador após remoção
    - Arquivo ficaria no sistema mesmo após dever ser removido
    - Atualiza contador de links
  - Link count muito baixo
    - Erro mais grave, provocaria remoção prematura do arquivo
    - Atualiza contador de links

Consistência: fsck

- Outras ocorrências suspeitas que podem ser reportadas
  - Diretórios com muitos arquivos (e.g. Mais de mil)
  - Permissões estranhas (e.g. 0007)
  - Arquivos em diretório de usuário mas pertencentes ao root e com setuid ligado

Consistência: fsck

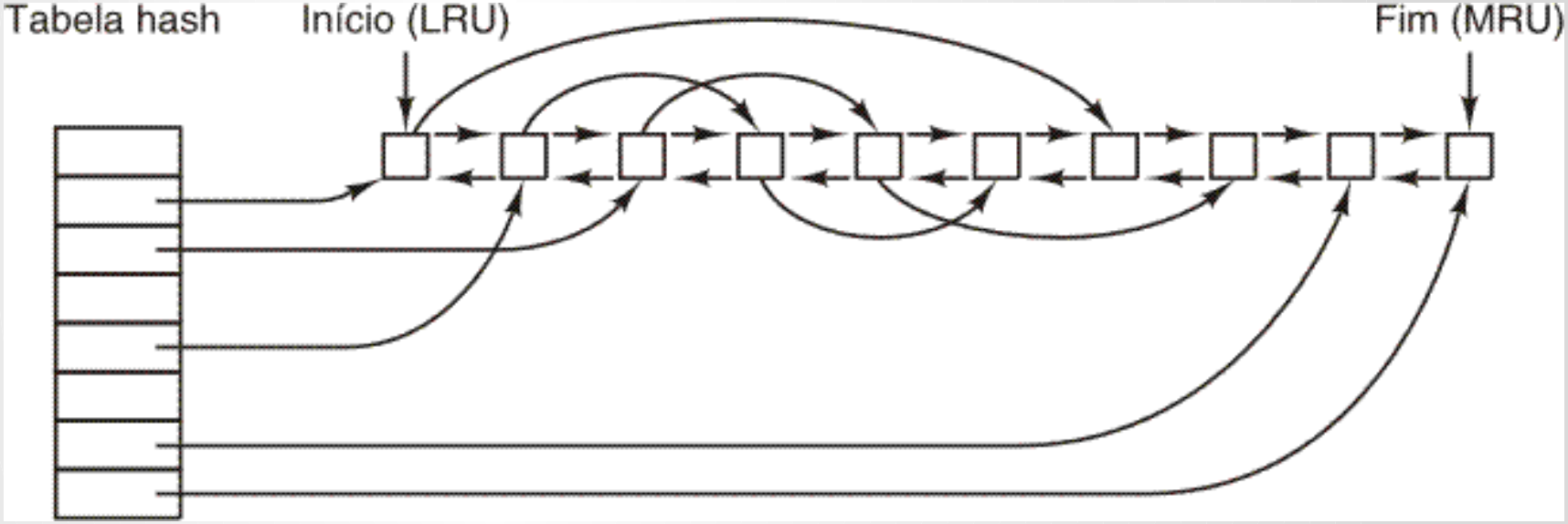
- Outras ocorrências suspeitas que podem ser reportadas
  - Diretórios com muitos arquivos (e.g. Mais de mil)
  - Permissões estranhas (e.g. 007)
    - Proprietário e grupo não tem acesso
    - Qualquer outro usuário tem! (?!)
  - Arquivos em diretório de usuário mas pertencentes ao root e com setuid ligado
    - Adquirem poderes de superusuário quando executados

DESEMPENHO DO SISTEMA DE ARQUIVOS

Performance do Sistema de Arquivos

- **Caches**
  - Blocos sempre carregados na área de cache antes de serem lidos
  - Tabela de hash (dispositivo+número do bloco) e lista ligada
  - Reposição semelhante aos algoritmos de memória virtual
    - Como caches lidas com menos frequencia (e sempre chamda de sistema) é viável o uso de LRU
    - Hash+ lista livre (quando bloco usado vai para fim)
  - Considerações adicionais
    - Blocos mais antigos em geral modificados
    - i-nodes são importantes para consistência do sisgtema

Performance do sistema de arquivos



Performance do Sistema de Arquivos

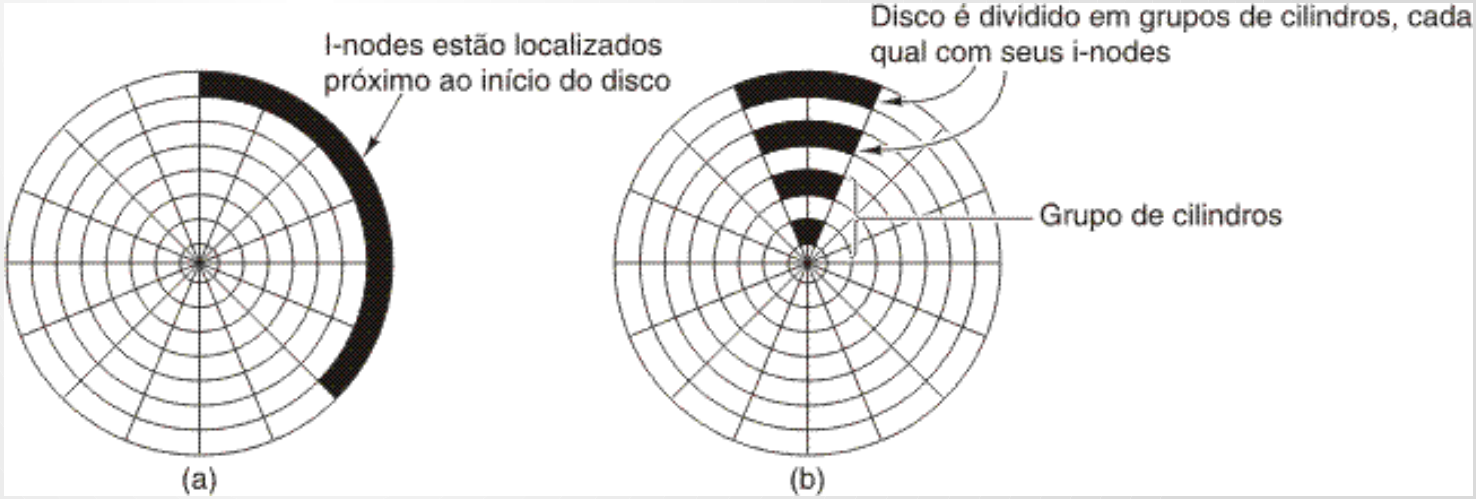
- **Caches**
  - Blocos devem ser divididos em categorias (i-nodes, blocos indretos, diretórios, blocos de dados complets, blocos de dados parciais)
    - Blocos que não devem ser usados tão cedo vão na frente, outros no final (como blocos que são de arquivos abertos para escrita e que estão parcialmente completos)
  - Blocos que são essenciais para consistência do sistema (i.e. Todos menos os blocos de dados)
    - Devem ser implementados write-trhough.
  - Mesmo assim, não deveríamos deixar blocos modificados sem serem escritos por muito tempo
    - UNIX – syncs periódicos (30 segundos?)
    - Windows – write-through cache. (i.e. USB drives – FAT – em geral são seguros)

Performance do Sistema de Arquivos

- **Leitura preventiva de blocos**
  - Tentar colocar blocos na cache antes de sua leitura/escrita ser requisitada
  - Ex. Maioria dos arquivos lido sequencialmente
    - FS pode ver, quando block K é requisitado, se bloco K+1 está lá, se não estiver, pode requisitar leitura preventiva
    - Como? – supor sequencial, quando seek é chamado, supor randomico (sem leitura preventiva),

Performance do Sistema de Arquivos

- Reduzindo movimento do braço do disco
  - Colocar blocos que podem ser utilizados sequencialmente próximos um do outro, de preferencia no mesmo cilindro
    - E.g. Se lista livre como bitmap, podemos pegar bloco o mais próximo possível do último escrito
    - Alocar mais de um bloco de cada vez para arquivo
    - I-nodes + arquivos: distribuir i-nodes por grupos de cilindros e alocar blocos dos arquivos preferencialmente próximos.
      - Variação – inodes no meio do disco



Proxima aula

- Sistemas de Entrada / Saída!

Referências

- TANENBAUM, Andrew S.. **Sistemas operacionais modernos**. 3. ed. São Paulo: Prentice Hall, 2009. 653 p. ISBN: 9788576052371.
  - **Capítulo 4**

- OLIVEIRA, Rômulo Silva de; CARISSIMI, Alexandre da Silva; TOSCANI, Simão Sirineo. **Sistemas operacionais**. 4. ed. Porto Alegre: Bookman, 2010. ISBN: 9788577805211.
  - **Capitulo 8**

- SILBERCHATZ, A.; Galvin, P.; Gagne, G.; **Fundamentos de Sistemas Operacionais**, LTC, 2015. ISBN: 9788521629399
  - **Capítulo 9**